APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/954,715	09/12/2001	Ping-Sheng Tseng	C5048AC04	8847

54698   7590   03/31/2008
RAYMOND R. MOSER JR., ESQ.
MOSER IP LAW GROUP
1030 BROAD STREET
2ND FLOOR
SHREWSBURY, NJ 07702

EXAMINER
SAXENA, AKASH

ART UNIT	PAPER NUMBER
2128	

MAIL DATE	DELIVERY MODE
03/31/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

United States Patent and Trademark Office

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Application Number: 09/954,715
Filing Date: September 12, 2001
Appellant(s): TSENG ET AL.

_____
Robert M. Brush
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 26th December 2007 appealing from the

Office action mailed 26th January 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

- U.S. Patent No. 5,663,900 issued to Bhandari et al (BH'900 hereafter)

- U.S. Patent No. 5,771,370 issued to Klein (Klein hereafter)

- U.S. Patent No. 5,661,662 issued to Michael R. Butts et al (BU '662 hereafter).

- IEEE article "Q-Modules: Internally clocked Delay-Insensitive Modules" by Fred U. Rosenberger et al (RO '1988 hereafter), publication September 1988.

- IEEE article "High Speed External Asynchronous/Internally clocked Systems" by W.S. VanScheik et al (VA '1997 hereafter) published July 1997.

- IEEE article "A Heterogeneous Environment for Hardware/Software Co-simulation" by William D. Bishop et al (BI '1997 hereafter), published 1997.

Supporting evidence used to provide common knowledge in the art of hardware and software co-simulation.

- U.S. patent 6,052,524

- U.S. Patent 5,546,562

- Structured Computer Organization; Andrew S Tanenbaum; Prentice Hall 1984; Pg.10-12 (Newly attached in view of arguments by applicant to support examiner's position and indicate what is common knowledge in art).

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

1. Claims 1-6, 19-36, 38 & 44-46 are rejected under 35 U.S.C. 103(a) as being

    unpatentable over U.S. Patent No. 5,663,900 issued to Narpat Bhandari et al

    (BH '900 hereafter), in view of U.S. Patent No. 5771370 issued to Klein (Klein

    hereafter).

Regarding Claim 1

BH '900 teaches an electronic design and automation (EDA) system for verification

and modeling a user design including a central processing unit (Sun Microsystems

Sparc workstation) (BH '900: Col.2, Lines 28-35; Col.1, Lines 14-18). Further, **BH**

'900 teaches an internal bus (SBus) system connected to the computing system (BH

'900: Col.2 Lines 46-50). Further, **BH '900** teaches a reconfigurable hardware logic

(BH '900: Col.2 Lines 64-67; Col.3 Lines 1-2) coupled to the internal bus system (BH

'900: Col.3 Lines 14-17, 21-24; Figure 2A-2B, Elements 16,38,44 & 46) for

generating a hardware model which includes at least a portion of user design

modeled in hardware (BH '900: Col.3 Lines 56-67; Col.4 Lines 1-2). Further, **BH**

'900 teaches a controller/interface circuit (BH '900: Figure 2B, Element 40), which on

a card directly attached on the motherboard (BH '900: Figure 2B, Element 38, 36)

between the external systems (BH '900: Figure 2B, Element 44-46) and computing

system. The controller mentioned above, controls the flow of control data,

synchronization data, initialization of logic states or simulation model data to the

external systems likes field programmable gate array (FPGA) (BH '900: Col.5 Lines

7-15). Further, **BH '900** teaches generating software models (BH '900: Col.1 Lines 25-28, Col.3 Lines 64-66) and hardware models (BH '900": Col.2 Lines 51-67; Col.3 Lines 1-2) and storing the models (BH '900: Col.3 Lines 2-5, 36-43).

BH'900 further teaches that the second information comprises at least one internal state of the hardware model (BH'900: Col.3 Lines 21-29). Examiner takes official notice that in view of arguments presented in the last office action that such information would be vital to the any hardware-software co-simulation system[1].

BH'900 **does not explicitly teach** a shared memory for holding first information of software model and second information of the hardware model, and the software model is capable of directly accessing the second information of the hardware model.

However, **BH'900** **explicitly teaches** that communication between the software model (prototype definition 32) and hardware model (on FPGA external system 44), and signal processing there-between. BH'900: Col.3 Lines 21-29 states:

> Interface software and hardware is provided between the simulated prototype definition 32 and external systems 44, 46 to provide distributed or multiple access paths for cooperative processing and signal processing therebetween. In this manner, specified signal paths or interconnection lines are provided, as specified by a design engineer, from CAE tool 14 to particular sockets pins or electrical contacts or nodes in particular external systems 44, 46 from particular simulator 16.

To emphasize that the limitation relating to shared resource (memory specifically), and direct access of the software model to the hardware model, although obvious in BH'900, teachings of **Klein** are presented below.

---

[1] See evidentiary support in US Pat. 6,052,524 Col.14 Lines 21-37 – Cited not used; U.S. Patent No. 5,546,562 – Abstract - Cited not used.

Klein teaches shared (coherent) memory (Klein: Fig.1 Element 22) for holding first information of software model (Klein: Fig.1 Element 34) and second information of the hardware model (Klein: Fig.1 Element 38), and the software model is capable of directly accessing the second information of the hardware model (Klein: Col.9 Fig.6 & related description; Col.11 Lines 48-49; Col.11 Line 63-Col.12 Line 9).

Here the first information is embodied as information present in the ISS 20, second information is embodied in the form of hardware model processor instance (Klein: Col.8 Lines 55-62).

Klein teaches the software model direct having direct access to the second information bypassing the hardware model (simulation) using the co-simulation optimization manager 27 (Klein: Col.11 Lines 48-49, Col.11 Line 63-Col.12 Line 9; Fig.10).This kind of memory access is by software model (ISS 20) is understood to be optimized access. Hardware accesses to the coherent memory are unoptimized accesses. (See Col.4 Lines 15-36).

It would have been obvious to one (e.g. a designer) of ordinary skill in the art at the time the invention was made to apply teachings of **Klein** to **BH '900** to have a coherent memory to solve the at speed simulation problem between the hardware and software models (Klein: Col.2 Lines 18-30, 47-53; BH'900: Col.1 Line 54-Col.2 Line 13). The motivation to combine would be that Klein see BH'900 model as one disclosed in Klein Col.2 Lines 18-20, as lacking speed, his invention enhances the performance of BH'900 hardware-software model by providing the shared memory concept with the co-simulation optimization manager 27.

Regarding Claim 2

BH '900 teaches first information of the software model including functional information of the user design (BH '900: Col.1 Lines 25-35).

Regarding Claim 3

BH '900 teaches second information of the hardware model including functional information of the user design (BH '900: Col.2 Lines 56-63).

Regarding Claim 4

BH '900 teaches hardware model includes the state information (BH '90: Col3. Lines 21-29, 51-55), which includes hardware node values and register-values (Specification: Pg.60 Line 26).

Regarding Claim 5

BH '900 teaches SBus configuration and the controller connected directly to the motherboard implementing the SBus (BH '900: Figure 2B Element 36-38; Col.2 Lines 46-50). DMA access is inherent to the SBus configuration (IEEE Std 1496-1993)[2].

Regarding Claim 6

BH '900 teaches hardware model includes the state information (BH '900: Col3. Lines 21-29, 51-55), which includes hardware node values and register-values (Specification: Pg.60 Line 26).

---

[2] IEEE Standard for Boot Firmware: Bus Supplement for IEEE 1496 (SBus). – 18th November 1994

Regarding Claim 19

Method claim 19 discloses the similar limitations as claim 1 and is rejected for the same reasons as claim 1. Further, **BH '900** teaches simulation method where the software model imitates functional or logical output signals (*first* information) in response to the stimulus applied (BH '900: Col.1 Lines 29-35).

Regarding Claim 20

Method claim 20 discloses the similar limitations as claim 5 and is rejected for the same reasons as claim 5. **_BH'900_** *further teaches that the third information asat least one internal state of the hardware model (BH'900:* Col.3 Lines 21-29*). Further, please see argument presented in the last office action that such information would be vital to the any hardware-software co-simulation system – see claim 1 evidentiary support.*

Regarding Claim 21

BH '900 teaches controlling the software and hardware model with a software kernel (BH '900: Col.2 Lines 7-13; Col.6 Lines 1-7).

Regarding Claim 22

BH '900 teaches software kernel includes model input and output, which is parsed and provided to the simulator through the PLI, thus allowing the simulation system to *determine* the presence of input data. **BH '900** teaches a digital simulator using Verilog; hence the *evaluation* of clock components[3] as part of the software program (test bench) is inherent to the method of simulation. Further, **BH '900** teaches

propagating the input data to the hardware model (BH '900: Fig. 3; Col.6 Lines 34-

62). Further, Examiner takes official notice that the step of detecting active clock

edge of the clock in the software model is well known in the art[4]. Further, **BH '900**

teaches evaluating the input data with the hardware model in response to the active

clock edge detection (BH '900: Col.4 Lines 3-9, Col.5 Lines 7-15).

Regarding Claim 23

BH '900 teaches simulating the behavior of the software model for some time and

then simulating the behavior of the circuit with hardware model for another time (BH

'900: Col.3 Lines 61-67; Col.4 Lines 1-2, 18-24).

Regarding Claim 24

Method claim 24 discloses the similar limitations as claim 1 and is rejected for the

same reasons as claim 1. Further, **BH '900** teaches simulating a behavior of the

circuit with the software model by providing input data to software model (BH '900:

Col.6 Lines 20-33). Further, **BH '900** teaches selectively switching to hardware

model through software control and providing input data (BH '900: Col.5 Lines 10-

15; 21-26). Further, **BH '900** teaches evaluating the input data in the hardware

model based on the trigger event in the software model (BH '900: Col.5 Lines 7-10).

Regarding Claim 25

BH '900 teaches hardware model includes the state information (BH '90: Col3. Lines

21-29, 51-55), and it is stored in the shared memory (BH '900: Col.3 Lines 36-43).

---

[3] IEEE Std 1364-1995 IEEE standard hardware description language based on the Verilog(R) hardware
description language. Cover Page & Pg. 101 with comments.
[4] IEEE Std 1364-1995: Cover Page & Pg. 370-371.

Regarding Claim 26

BH '900 teaches simulating the software model by using the state information (BH '900: Col.1 Lines 29-35) from hardware model in the shared memory (BH '900: Col.3 Lines 2-5, 36-43). Amendment to the claim does not change the response as the behavior of the circuit is the characteristic that being simulated by BH '900.

Regarding Claim 27

BH '900 teaches generating the hardware model further comprises of determining component type in hardware language and generating model based on the components (BH '900: Col.2 Lines 55-63; Col.3 Lines 51-55).

Regarding Claim 28

BH '900 teaches that portion of the user design can be simulated (in software) and portion of the user design can be emulated (in hardware) (BH '900:Col.3 Lines 61-64). Further, **BH '900** teaches selectively switching between hardware and software models (BH '900:Col.4 Lines 3-6). Further, **BH '900** teaches simulating the behavior of the circuit by providing input data to software model through the programmable logic interface (PLI) (BH '900: Figure 2A Elements 28, 30, 16).

Regarding Claim 29

BH '900 teaches software kernel includes model input and output, which is parsed and provided to the simulator through the PLI, thus allowing the simulation system to determine the presence of input data. **BH '900** teaches a digital simulator using Verilog; hence the evaluation of clock components (See Footnote 2 reference) as part of the software program (test bench) is inherent to the method of simulation.

Further, **BH '900** teaches propagating the input data to the hardware model (BH
'900: Fig. 3; Col.6 Lines 34-62).

Regarding Claim 30

Claim 30 discloses the similar limitations as claim 1 and is rejected for the same
reasons as claim 1. Claim 1 discloses steps of generating the software and
hardware models, allocating shared memory. Further, **BH '900** teaches propagating
data to the hardware model until data stabilizes as calculating the propagation delay
as part of the initialization setup (BH '900: Col.5 Lines 27-30). Further, **BH '900**
teaches detecting the clock edge being implicit to the Verilog model/test bench (See
IEEE Std 1364-1995). Further, **BH '900** teaches evaluating data with the hardware
model in response to the clock edge detection in software model and in
synchronization with a hardware-generated clock by external system/FPGA (BH
'900: Col. Lines 64-67).

Regarding Claim 31

Claim 31 discloses the similar limitations as claim 6 and is rejected for the same
reasons as claim 6.

Regarding Claim 32

BH '900 teaches simulating the software model by using the state information (BH
'900: Col.1 Lines 29-35) from hardware model in the shared memory (BH '900: Col.3
Lines 2-5, 36-43).

Regarding Claim 33

BH '900 teaches a simulation system operating in a host system for simulating the
behavior of a circuit (BH '900: Figure 2A-2B) containing CPU (BH '900: Col.2 Lines
51-55), shared memory (BH '900: Col.3 Lines 2-5, 36-43) and local bus coupling the
CPU to memory (BH '900: Col.2 Lines 46-50). BH '900 teaches a circuit having
structure and function in a hardware language capable of describing the component
types and connection (BH '900: Col.2 Lines 56-63; Col.3 Lines 48-55).

Further, BH '900 teaches software model coupled to the local bus (BH '900: Figure
2A Elements 16-34-36), software control logic (BH '900: Figure 2A Element 30)
coupled to the software model & hardware logic element (BH '900: Figure 2A-2B
Elements 38, 40, 42-46) for controlling the operation of software model and
hardware logic element. Further, BH '900 teaches hardware logic element coupled
to local bus (BH '900: Figure 2B Elements 36-38) and hardware model including at
least a portion of the circuit (BH '900: Col.3 Lines 61-67; Col.4 Lines 1-2) configured
automatically based on component type (BH '900: Col.3 Lines 48-54).

Further, BH '900 teaches SBus configuration and the controller connected directly to
the motherboard implementing the SBus (BH '900: Figure 2B Element 36-38; Col.2
Lines 46-50). DMA engine based access is inherent to the SBus configuration.

BH'900 further teaches that the second information comprises at least one internal
state of the hardware model (BH'900: Col.3 Lines 21-29). Further, please see
argument presented in the last office action that such information would be vital to
the any hardware-software co-simulation system.

BH'900 further teaches the software model is capable of directly accessing the second information of the hardware model (Klein: Col.9 Fig.6 & related description; Col.11 Lines 48-49; Col.11 Line 63-Col.12 Line 9).

Regarding Claim 34 & 35

Claims 34 & 35 discloses the similar limitations as claim 21 & 22 and are rejected for the same reasons as claims 21 & 22.

Regarding Claim 36

BH '900 teaches hardware logic element comprises a field programmable device (BH '900: Col.3 Lines 1-2).

Regarding Claim 38

Claim 38 discloses the similar limitations as claim 1 and is rejected for the same reasons as claim 1. Further, BH '900 teaches control logic coupled to internal bus system for delivering the data among reconfigurable hardware logic and computing system (BH '900: Fig. 3; Col.6 Lines 34-62).

Regarding Claim 44

Claim 44 discloses the similar limitations as claim 1 and is rejected for the same reasons as claim 1.

Regarding Claim 45

BH '900 teaches synchronizing the data evaluation in the software model and hardware model with software configured/generated clock (BH '900: Col.5 Lines 64-67).

Regarding Claim 46

BH '900 teaches simulating selected debug test points in the software as software
program (kernel) which can control the simulation flow by starting, stopping, single-
stepping, interrupting, or polling the simulation (BH '900: Col.6 Lines 5-7).
Further, BH '900 teaches accelerating selected portions in hardware (BH '900: Col.3
Lines 61-67, Col.4 Lines 1-2). Further, BH '900 teaches controlling the delivery of
data between the hardware and software model through the external I/O so that
software model has access to all delivered data (BH '900: Col.4 Lines 39-46).

2. **Claim 7 & 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over
   U.S. Patent No. 5,663,900 issued to Narpat Bhandari et al (BH '900 hereafter),
   in view of U.S. Patent No. 5771370 issued to Klein (Klein hereafter), further in
   view of IEEE article "Q-Modules: Internally clocked Delay-Insensitive Modules"
   by Fred U. Rosenberger et al (RO '1988 hereafter).**

Regarding Claim 7

Teachings of **BH '900** are disclosed on claim 1 rejections above.

   BH '900 does not teach timing logic associated with each flip flop so the desired
result is obtained regardless of the order of arrival of a input and a clock signal to the
flip flop.

   RO '1988 teaches their Q-modules are insensitive to delay and that the inputs
must be able to accept inputs that change at any time with respect to the clock and
must operate correctly in the presence of metastability (RO '1988: Pg.1006 Section
II; Pg 1009 Section III D).

Motivation to combine **Klein** with **BH'900** is provided in claim 1 rejection above.

It would have been obvious to one (e.g. a designer) of ordinary skill in the art at the time the invention was made to apply teachings of **RO '1988** to **BH '900** to make logic elements that are delay insensitive. The motivation to combine would have been that **BH '900** is designing a hardware emulated and software simulated co simulation system, where parasitic effects of propagation delays between the hardware and software boundaries are of concern (**BH '900**: Col.5 Lines 27-30) and **RO '1988** solves this problem by teaching that Q-modules which can be at the interface of simulation environment and FPGA hardware boundary where such metastable transitions are most possible (RO '1988: Pg. 1005 Last paragraph). Further, motivation would have been that **RO '1988** discloses that Q-modules are used in the personalizing the PLAs (Abstract).

Regarding Claim 10

Claim 10 discloses the similar limitations as claim 7 and is rejected for the same reasons as claim 7.

3. **Claims 8-9 & 11-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 5,663,900 issued to Narpat Bhandari et al (BH '900 hereafter), in view of U.S. Patent No. 5771370 issued to Klein (Klein hereafter), further in view of IEEE article "Q-Modules: Internally clocked Delay-Insensitive Modules" by Fred U. Rosenberger et al (RO '1988 hereafter), further in view of IEEE article "High Speed External Asynchronous/Internally clocked Systems" by W.S. VanScheik et al (VA '1997 hereafter).**

Regarding Claim 8 & 9

Teachings of **BH '900** & **RO '1988** are disclosed on claim 1 rejections above.

BH '900 does not teach timing logic having a first logic, second logic, third logic and edge detector.

RO '1988 also does not teach the exactly the disclosed limitations for the first logic, second logic, third logic and edge detector. However the solve the same problem as disclosed by the limitations.

VA '1997 teaches a first logic (VA '1997: Fig.5 Input Registers & Next State Logic) having first input (VA '1997: Fig 5 INPUTS), a second input (VA '1997: Fig.5 Second feedback input to Next State Logic), a control input (VA '1997: Fig.5 Clock signal to the Input registers) and an output (VA '1997: Fig.5 output of Next State Logic). Further, **VA '1997** teaches a second logic (VA '1997: Fig.5 Memory Registers) with first trigger input (VA '1997: Fig.5 Clock) and first logic output coupled to the second input of the first logic (VA '1997: Fig.5 Memory register output to the next state logic) where second logic updated the first logic. Further, **VA '1997**

teaches a third logic storing the new value (VA '1997: Fig.5: Output Logic). Further,

VA '1997 teaches a edge detector having a third trigger and clock input and output

is coupled to the control input (VA '1997: Fig.5: Majority Gate & Clock Driver).

The limitation "first trigger input" is taught by the prior art (VA '1997: Fig.5 Clock -

first trigger input). _The limitation "new value of first data" provided in claim 9 is still_

taught by the prior art in as a third logic storing the new value (VA '1997: Fig.5:

Output Logic). Further, examiner takes official notice that edge detectors are well

known in the art[5].

    Motivation to combine **Klein** with **BH'900** is provided in claim 1 rejection above.

    It would have been obvious to one (e.g. a designer) of ordinary skill in the art at

the time the invention was made to apply teachings of **RO '1988** to **BH '900** to make

logic elements that are delay insensitive. The motivation to combine would have

been that **BH '900** is designing a hardware emulated and software simulated co

simulation system, where parasitic effects of propagation delays between the

hardware and software boundaries are of concern (**BH '900**: Col.5 Lines 27-30) and

RO '1988 solves this problem by teaching that Q-modules which can be at the

interface of simulation environment and FPGA hardware boundary where such

metastable transitions are most possible (RO '1988: Pg. 1005 Last paragraph).

    Further, It would have been obvious to one (e.g. a designer) of ordinary skill in

the art at the time the invention was made to apply teachings of **VA '1997** to **BH**

'900 to make logic elements that are delay insensitive. The motivation would have

---

[5] U.S. Patent No. 5808486, Figure 1

been that **VA '1997** further refines the work of **RO '1988** (VA '1997: Pg824, Col.2

Lines 1-2) and handles the propagation delay issues across asynchronous

interfaces like in **BH '900** (VA '1997: Abstract: Lines 1-4) to avoid metastability.

Regarding Claim 11

VA '1997 teaches an timing circuit with input logic (VA '1997: Fig.5 Input Registers &

Next State Logic) for receiving the input value and trigger signal (VA '1997: Fig 5

INPUTS, Clock), a storage logic (VA '1997: Fig.5 Memory Registers), a transmission

logic (VA '1997: Fig.5: Output Logic) coupled to the input logic and storage logic for

selecting between the new and old values and outputting it. Further, **VA '1997**

teaches an edge detecting logic (VA '1997: Fig.5: Majority Gate & Clock Driver).

VA '1997 teaches a selection logic (VA '1997: Fig.5: Output Logic) coupled to the

input logic and storage logic for selecting between the new and old values and

outputting it.

Regarding Claims 12-18

The limitations of claims 12-18 recite "D-Flip Flop", multiplexers, AND gates, OR

gates etc. The examiner has interpreted these elements to be functionally equivalent

to the circuit disclosed by VA '1997 (VA '1997: Fig.3, 5 & 6). *VA '1997 teaches a*

selection logic (VA '1997: Fig.5: Output Logic) coupled to the input logic and storage

logic for selecting between the new and old values and outputting it.

4. **Claim 37 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S.**

    Patent No. 5,663,900 issued to Narpat Bhandari et al (<u>BH '900</u> hereafter) in view

    of U.S. Patent No. 5771370 issued to Klein (<u>Klein</u> hereafter), further in view of

    U.S. Patent No. 5,661,662 issued to Michael R. Butts et al (<u>BU '662</u> hereafter).

<u>Regarding Claim 37</u>

Teachings of **BH '900** are disclosed on claim 33 rejections above.

 BH '900 does not teach plurality of field programmable devices coupled together

separable by at most two interconnection.

 BU '662 teaches plurality of field programmable devices coupled together (BU

'662: Figure 3) separable by at most two interconnection (BU '662: Figure 7; Col.11

Lines 33-43).

 Motivation to combine <u>**Klein** with **BH'900**</u> is provided in claim 1 rejection above.

 It would have been obvious to one (e.g. a designer) of ordinary skill in the art at

the time the invention was made to apply teachings of **BU '662** to **BH '900** to make a

multi FPGA system coupled together. The motivation would have been, as **BU '662**

teaches, generally it takes more than one FPGA to implement the desired design

(BU '662: Col.3 Lines 3-31) for any practical system. Hence **BH '900** design would

also require more than one FPGA to implement the design and **BU '662** teaches

how to implement a multiple FPGA design while handling the interconnect issue

through the Realizer technology by Quickturn.

5.   Claims 39-43 & 47-50 are rejected under 35 U.S.C. 103(a) as being

unpatentable over U.S. Patent No. 5,663,900 issued to Narpat Bhandari et al

(BH '900 hereafter) in view of U.S. Patent No. 5771370 issued to Klein (Klein

hereafter), further in view of IEEE article "A Heterogeneous Environment for

Hardware/Software Cosimulation" by William D. Bishop et al (BI '1997

hereafter).

Regarding Claim 39
Claim Interpretation:

"Data-in pointer logic" and "Data-in latch logic" are together interpreted as hybrid

decoder and cross-point router. "Data-in pointer logic" is generating the select signal

based on the source of data present on the internal bus and "Data-in latch logic"

routes the data based on the select signal to the selective internal nodes in the

reconfigurable hardware.

Teachings of BH '900 are disclosed on claim 38 rejections above.

BH '900 does not teach plurality of field programmable devices "Data-in pointer

logic" and "Data-in latch logic" with the disclosed limitations.

BI '1997 teaches an interface platform (BI '1997: Pg.15 Section 2.3; 4.2) that

performs the function of directing the data from the internal data bus to the various

FPGA's based on the external interface or computing system (BI '1997: Fig.1, 3).

For example: The external interface may be the development platform and the

computing system may be the software simulation platform. Further, the limitations

"Data-in pointer logic" and "Data-in latch logic" disclosed are obvious by necessity in

the described interface.

Motivation to combine **<u>Klein</u>** with **BH'900** is provided in claim 1 rejection above.

It would have been obvious to one (e.g. a designer) of ordinary skill in the art at

the time the invention was made to apply teachings of **BI '1997** to **BH '900** to design

an interface control logic. The motivation would have been that **BI '1997** is

addressing the same issues as **BH '900** to perform hardware/software co-simulation

using the Sun Sparc station and FPGA's (BI '1997: Figure1, 2). Further, **BH '900**

also discloses a pin arrangement for the interface circuit between the hardware and

software (BH '900: Fig.3).

Regarding Claim 40

BI '1997 teaches an external buffer for storing the external data from the external

interface so the data is mutually accessible by buffer and the computing system (BI

'1997: Pg.18 Col.2 6th paragraph; Section 4.2 1st paragraph).

Regarding Claim 41

Claim 40 recites the similar limitation as claim 39 but data is transferred in the

opposite direction. **BI '1997** teaches a bidirectional interface to monitor and seed the

simulation (BI '1997: Fig.2) as well as configures the FPGA's (BI '1997: Pg. 15 Col.1

Last paragraph).

Regarding Claim 42

BI '1997 teaches control and data signal being sent back between the software

simulation and the hardware emulation (BI '1997: Fig.2). Further, detection of

software clock and evaluation in well known in the art and obvious by necessity of the current co-simulation design (See IEEE Std 1364-1995 reference cited before).

Regarding Claim 43

BI '1997 teaches selecting which components of the design as hardware and software entities and describes the reasoning why one would select a component to be simulated rather than being emulated (BI '1997: Section 3). It is obvious from the discussion the reasons why some of the external I/O would be simulated rather being emulated (i.e. emulation is performed for components where timing & implementation details are necessary and simulation is performed where functionality needs to be checked).

Regarding Claim 47

Claim 47 discloses the similar & subset of limitations as claim 39 and is rejected for the same reasons as claim 39.

Regarding Claim 48

Claim 48 discloses the similar limitations as claim 40 and is rejected for the same reasons as claim 40.

Regarding Claim 49

Claim 49 discloses the similar limitations as claim 39 and is rejected for the same reasons as claim 39.

Regarding Claim 50

Claim 50 discloses the similar & subset of limitations as claim 43 and is rejected for the same reasons as claim 43.

(10) Response to Argument

I Response to rejection for claims 1-6, 19-36, 38 & 44-46 unpatentable over BH'900 in view Klein.

(Argument 1) Applicant has argued in Remarks Pg.11:

> **First**, Appellants traverse the Examiner's official notice. It is not common knowledge *[sic]* well-known in the art that information comprising at least one internal state of the hardware model **[1]** is vital to any hardware-software co-simulation system. In support of the official notice, the Examiner cited U.S. patent 6,052,524, col. 14, lines 21-37, which states:
>
> > "Co-verification simulators known in *art **typically*** determine the *internal behavior* and *state of hardware and software components every cycle of a simulator*. **[3]** This provides accuracy, but may slow the simulation by an order of magnitude or more. **[5]** Cycle-accurate simulator 12 in a preferred embodiment of the present invention *allows events to be set and the internal behavior and state of hardware components to be accurately examined at specific events* **[4]** (e.g., after a memory access). This allows cycle-accurate simulator 12 to provide faster simulations of hardware and software components. *Cycle-accurate simulator 12 can also determine internal behavior and state of hardware and software components every cycle* by setting a first event to occur on a first simulator cycle and a second event to occur on a second simulator cycle, a third event to occur on a third simulator cycle, etc."

Applicant has further stated:

> Nothing in the cited passage, however, indicates that information comprising at least one internal state of the hardware model is "vital" to a co-simulation system. The passage states that co-verification systems "typically" determine internal behavior and state of hardware and software components. "Typically" does not mean vital or necessary, but rather indicates that in some cases co-verification systems do not determine internal behavior and state of hardware and software components. **[2]**

(Response 1) As per [1] and [2], examiner had provided two U.S. Patents to support the common knowledge in the art for obviousness rejection. Although the U.S. Patent No. 6,052,524, does not use the word "vital", the argument made by applicant seems to appear, if the information is "typically" known and used the art of hardware & software co-simulation, it is not common knowledge *and* well-known in the art. Further, applicant's statement ""Typically" does not mean vital or necessary, but rather indicates that in some cases co-verification systems **do not determine** internal behavior and state of hardware and software components." is incorrect as it shows that in most cases co-verification systems **do**

determine <u>internal behavior and state of hardware and software components</u>. Applicant has

merely alleged the above statement. Examiner respectfully disagrees with the

applicant.

(Argument 2) Applicant has argued in Remarks Pg.11-12:

> Moreover, <u>"determining" internal behavior and state of hardware and software components does</u>
> <u>not teach or suggest storing them in a shared memory</u>, as recited in Applicants' claim 1.
>
> **The Examiner also cited the Abstract of U.S. patent 5,546,562,** which states:
> << Disclosure of the patent from Remarks Pg.11-12>>

(Response 2) Examiner respectfully disagrees as shared memory is also shown as

well known in the art of hardware software co-simulation.

U.S. Patent 5,546,562 specifically states:

> "… <u>One access is from a **host simulation environment (26)** while the **other is from a**</u>
> <u>**model debug user interface (20)** where ***internal architecturally visible registers and***</u>
> <u>***status*** are available to the user for greater debug control on the simulated subsystem</u>
> <u>within simulation environment (26).</u> **[1]** Specifically, any of a wide variety of physical VLSI
> circuits (48) to be modeled is kept in a quiescent state after power-on by a device control
> (50). <u>It is then accessed through simulation means by simulated subsystem within a</u>
> <u>simulation environment (26), to change the architecturally visible internal state of the</u>
> <u>VLSI circuit (48). Control (50) brings VLSI circuit (48) out of the quiescent state and</u>
> <u>submits the requested simulated access. After taking the response, control (50) returns</u>
> <u>VLSI circuit (48) again to its quiescent state so as to keep its internal state current.</u> **[2]**
> The response is sent back to simulation environment (26) to update the simulated
> subsystem. Independently, any user request for accessing the architecturally visible
> internal state of the circuit is gathered by model debug and user interface (20). Interface
> (20) enables control (50) to bring VLSI circuit (48) out of the quiescent state and to
> submit the user request access. …"

The underlined disclosure above clearly shows the registers (which are storage

devices making the memory) are visible and available for debugging from the two

environments (26 and 20).

Examiner has shown two examples (U.S. Patent No. 6052524 and 5546562) where

not only hardware software co-simulation uses sharing at least one internal state

during co-simulation (U.S. Patent No. 6052524 see [3] & [4] in the Argument 1

section), but also having this information residing on the shared memory (registers)

(U.S. Patent No. 5546562 see [1] & [2] in the Response 2 section). Applicant has

merely alleged without support as this not being common knowledge at the time of

invention.

(Argument 3) Applicant has argued in Remarks Pg.12:

> The Examiner further stated that "[i]t is well known in the art that co-simulation of hardware and software share information, otherwise it would not be called co-simulation." (Final Office Action, pg.3). Appellants, however, do not claim a general exchange of information between hardware and software. Rather, Appellants recite that the second information comprises <u>at least one internal state of the hardware model</u>. In essence, Appellants claim a shared memory for storing first information of a software model and the second information, which allows the software model to directly access at least one internal state of the hardware model. <u>The mere fact that co-simulation generally involves sharing information between hardware and software does not teach, suggest, or otherwise render obvious the more specific feature of a shared memory configured to store both information of the software model and internal state information of the hardware model, recited in Applicants' claims.</u>

(Response 3) Examiner has clearly shown in U.S. Patent No. 5546562 (see [1] & [2]

in the Response 2 section) that internal state is on the register which is shared

memory.

(Argument 4) Applicant has argued in Remarks Pg.13:

> **Second**, the Examiner has not set forth a prima facie case of obviousness with respect to BH '900 alone. The Examiner stated "[t]o emphasize that the limitation relating to shared resource (memory specifically), and direct access of the software model to the hardware model, although obvious in BH '900 .... " This is a conclusory statement and does not set forth any rationale for why access between a software model and a hardware model teaches, suggests, or otherwise renders obvious a shared memory, as recited in Appellants' claims.

(Response 4) Applicant has not cited the full sentence form the rejection and

misread that section as motivation to combine. The complete sentence and full

motivation to form the obviousness argument is presented below.

> To emphasize that the limitation relating to shared resource (memory specifically), and direct access of the software model to the hardware model, although obvious in BH'900, teachings of **Klein** are presented below.

The teaching of Klein was added to clearly teach the limitation although present in

BH'900. This is not a conclusory statement. The complete motivation to combine

was provided at the end of rejection as follows:

> It would have been obvious to one (e.g. a designer) of ordinary skill in the art at the time the
> invention was made to apply teachings of **Klein** to **BH '900** <u>to have a coherent memory to solve
> the at speed simulation problem</u> between the hardware and software models (Klein: Col.2 Lines
> 18-30, 47-53; BH'900: Col.1 Line 54-Col.2 Line 13). **The motivation to combine** would be that
> Klein see BH'900 model as one disclosed in Klein Col.2 Lines 18-20, <u>as lacking speed, his
> invention enhances the performance of BH'900 hardware-software model</u> by providing the shared
> memory concept with the co-simulation optimization manager 27.

In view of recent ruling from KSR international vs. Teleflex Inc. at least the following

rationales apply in addition to the motivation to combine.

> (C) Use of known techniques to improve similar device (method or product) in the same way and
> (D) Applying a known technique to a known device (method or product) ready for improvement to yield
> the predictable result.
> In this case U.S. Patents No. 6,052,524 and 5,546,562 show hardware software co-simulation
> techniques similar to the claimed invention where internal states are accessed through shared
> memory (registers).
>
> (E) "Obvious to try" - choosing from a finite number of identified, predictable solutions, with reasonable
> expectation of success.
> BH'900 and Klein motivation to combine speeds the hardware-software co-simulation. This is a
> known problem which is also illustrated by U.S. Patents No. 6,052,524 (See Argument 1 point [5]
> above). It would be obvious to try to speed the simulation for achieve results faster.
>
> (G) The analysis and presentation of teaching, suggestion and motivation to combine is same as done inline
> with MPEP 2143.01 below in the rejection.
> The analysis is presented above and in the claim rejection.

(Argument 5) Applicant has argued in Remarks Pg.13:

> **Third**, Klein discloses co-simulation of a hardware-software system design. (See Klein, Abstract).
> That is, a hardware portion of a design is simulated along with a software portion of the design.
> The software portion of the design is simulated using an instruction set simulator (ISS). The
> hardware portion of the design is simulated using a logic simulator in conjunction with software-
> based processor and memory models. (Klein, col. 5, lines 6-65; FIG. 2). Both the logic simulator
> and the ISS are software-based simulators. (Klein, col. 5, lines 43-50; col. 2, lines 6-8). Klein
> states that "it should be noted that the present invention may be practiced in one or more [sic]
> generate or special purpose computer systems." (Klein, col. 6, lines 38-40).
>
> Thus, Klein and Appellants' invention are directed to two different types of simulation. In Klein, a
> circuit design comprising both hardware and software portions (sometimes referred to as an
> embedded system - see Klein, co1.1, lines 34-65) is simulated using a computer system. <u>In
> Appellants' invention, a circuit design is simulated using a computer system coupled to a
> reconfigurable hardware system (e.g., FPGAs).</u> "Co-simulation" in Klein refers to the

simultaneous simulation of hardware and software portions of a circuit design on a computer system. In the context of Appellants' invention, "co-simulation" means simulation of a circuit design (whether an embedded system or not) using both a computer and reconfigurable hardware. Klein is completely devoid of any teaching or suggestion of a reconfigurable hardware system for simulation, emulation, hardware acceleration, etc. of a hardware model.

(Response 5) Applicant has pointed that (hardware) logic simulator and (software)

Instruction set simulator (ISS) are both software based simulator as compared to the

claimed "reconfigurable hardware logic" to simulate the hardware design. BH'900 is

used to teach this limitation (BH'900: Fig.1 and 2A-B). Klein is used to teach details

shared memory between the hardware logic simulator and software simulator (Klein:

Col.11 Lines 48-49, Col.11 Line 63-Col.12 Line 9; Fig.10). Hence applicant is

performing piecemeal analysis of the combination.

Further, even if the hardware logic simulator, which models the hardware design,

is not implemented in hardware, like an FPGA as claimed, it is known in the art that

the "hardware and software are logical equivalent"[6] (Tanenbaum 1984). Tanenbaum

further teaches having program on FPGA is not a novel concept.

An intermediate form between hardware and software is **firmware**, which consists of software embedded in electronic devices during their manufacture. Firmware is used when the programs are rarely or never expected to be changed, for example, in toys or appliances. Firmware is also used, even when the programs must not be lost when the power is off (e.g., when the doll's battery runs down). In many computers, the microprogram is is firmware.

A central theme of this book that will occur over and over again is:

Hardware and software are logically equivalent.

Any operation performed by software can also be built directly into the hardware and any instruction executed by the hardware can also be simulated in software. The decision to put certain functions in hardware and others in software is based on such factors as cost, speed, reliability, and frequency of expected changes. There are no hard and fast rules to the effect that X must go into the hardware and Y must be programmed explicitly. Designers with different goals may, and often do, make different decisions.

---
[6] Structured Computer Organization; Andrew S Tanenbaum; Prentice Hall 1984; Pg.10-12

(Argument 6) Applicant has argued in Remarks Pg.:

> Thus, Klein does not teach or suggest any hardware component (i.e., a shared memory) that can
> be used in the hardware portion of BH '900 to emulate a portion of a design. Accordingly, Klein
> does not bridge the substantial gap between BH '900 and Appellants' invention. Namely, Klein
> does not provide a teaching or suggestion of a shared memory that can be incorporated into or
> otherwise modify the system of BH '900. Therefore, Appellants contend that no conceivable
> combination of BH '900 and Klein renders obvious Appellants' invention.

(Response 6) Klein teaches shared memory as coherent view of the memory (Klein:

Fig.1). Further the teaching of Klein supplements BH'900 in providing details of the

coherent view of the memory (Klein: Fig.2 & 4). Klein is not concerned with exactly

how the details of the hardware simulator/emulator and software simulator are

implemented (Klein: Col.5 Lines 41-65), which is taught by BH'900 (BH'900: Fig.1

and 2A-B).

(Argument 7) Applicant has argued in Remarks Pg.14:

> **Fourth**, the software described in Klein performs the hardware and software simulations with a
> single coherent view of the memory of the hardware-software system being co-simulated. (Klein,
> col. 4, lines 15-21). This "single coherent view of memory" in Klein is not a shared memory, as
> recited in Appellants' claims. Appellants clearly claim a physical shared memory, as the shared
> memory is included in a system having a computing system, an internal bus, reconfigurable
> hardware logic, and control logic. **[1]** One skilled in the art would clearly understand Appellants'
> shared memory to be physical in the context of the claims and in light of Appellants' specification.
> Appellants contend that to interpret the claims otherwise is unreasonable. Appellants are not
> required to preface each element in a claim with the term "physical."
> In contrast to Appellants' invention, the "single coherent view of memory" in Klein is an abstract or
> logical construct of a memory portion of the hardware-software design being simulated. The
> "memory" referred to in Klein is not the memory of the co-simulation system, but is rather the
> memory portion of the design being co-simulated. **[2]** The Examiner's cite to FIG. 1 of Klein
> actually confirms this teaching. (Final Office Action, p. 5). FIG. 1 of Klein recites "memory of
> hardware-software system ~ co-simulated." The plain meaning of that statement is that the
> memory is part of the hardware-software system, and that such hardware-software system is
> being co-simulated. There is no contrary meaning in Klein. This logical memory construct
> described in Klein does not add anything to the communication of physical signals between the
> simulator and the external systems in BH '900.

(Response 7) The claim 1 recites:

> "... **shared memory** for holding a first information of a software model and a second information
> of the hardware model, where the second information comprises at least one internal state of the
> hardware model and the software model is capable of directly accessing the second information of
> the hardware model."

Applicant's arguments as in [1] and [2] above require a specific reading. However the claims do not recite an invention which requires a specific reading, therefore a broad interpretation of the claimed limitations is made here. Specifically, as per [1] claim does not recite a physical shared memory and as per [2], claim does not differentiate between the logical and physical memory as argued.

For at least the reasons above examiner finds applicant's arguments unpersuasive and requests boards affirmation.

II Response to rejection for claims 7 & 10 unpatentable over BH'900 and Klein and RO'1988.

No new arguments are presented for this rejection. Arguments presented above are incorporated herein. Teachings and motivation to combine with additional references is presented in the Grounds of Rejection section earlier.

III Response to rejection for claims 8-9 & 11-18 unpatentable over BH'900, Klein, RO'1988 and VA'1997.

No new arguments are presented for this rejection. Arguments presented above are incorporated herein. Teachings and motivation to combine with additional references is presented in the Grounds of Rejection section earlier.

IV Response to rejection for claim 37 unpatentable over BH'900, Klein and Butts.

No new arguments are presented for this rejection. Arguments presented above are incorporated herein. Teachings and motivation to combine with additional references is presented in the Grounds of Rejection section earlier.

V Response to rejection for claims 39-43 & 47-50 unpatentable over BH'900, Klein and BI'1997.

No new arguments are presented for this rejection. Arguments presented above are incorporated herein. Teachings and motivation to combine with additional references is presented in the Grounds of Rejection section earlier.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Akash Saxena/

GAU 2128, Friday, March 14, 2008

Conferees:

/Kamini S Shah/
Supervisory Patent Examiner, Art Unit 2128

Kamini S. Shah
Supervisory Patent Examiner, GAU 2128
Structural Design, Modeling, Simulation and Emulation

/Eddie C Lee/
Supervisory Patent Examiner, TC 2100

Eddie Lee
Supervisory Patent Examiner